COVID-19 DIAGNOSIS USING NEURAL NETWORKS FOR IMAGE CLASSIFICATION

Ananya A. , Gargi V., Keerat S., Muzeeb S., Mythili N.

May 5, 2020

1 Introduction

With the ongoing coronavirus pandemic, diagnostic tests have been urged to be an essential tool to track and contain the spread of the disease. Most testing is done on viral genetic material from throat and nose swabs using a highly specific PCR (polymerase chain reaction) test. These typically take upto 24 hours in testing time and are very limited in supply. On the other hand, a chest X-ray or a CT scan, which is available within few hours, can provide a preliminary diagnosis of the disease. A PCR test, though very specific, has low sensitivity, which means a test can be negative even when the patient is infected. Comparatively, a chest X-ray scan is a highly sensitive test, with lower specificity for COVID-19, since other respiratory infectious diseases like pneumonia share similar characteristic features of infection in an X-ray. Technology can play a crucial role to help supplement the diagnostic centers to cope with the daily growing number of people showing symptoms to the disease. Artificial intelligence techniques can help automate the whole process of COVID-19 diagnosis, without requiring a domain expert and increase the capacity of testing at medical centers at remote locations as well.

In this project, we thus aim at developing a fully automatic framework for the diagnosis of respiratory infectious diseases, specifically a COVID-19 infection, using a chest X-ray scan. We have compared different machine learning approaches and implemented various optimization techniques for selecting the best model. Additionally, we have also developed a user interface for our diagnostic tool that can help in a fast, scalable and affordable diagnostic test for the SARS-CoV-2. With these goals we have explored on a promising approach towards medical diagnosis using convolutional neural networks for X-ray Image Classification.

2 Link to Repository

https://github.tamu.edu/an-agrawal/CSCE633_S20_Ananya_Gargi_Keerat_Muzeeb_Mythili_Project.git

3 Related Work

The success of Deep learning algorithms for image segmentation, localization, classification, and recognition task in recent years is timely with remarkable increase in medical image data [17]. Analysis of these data has become an active field, partly as image data are easier for clinicians to interpret and they are relatively structured and labeled. There have been reported accuracies in some publications for detecting a range of anomalies such as malignant tumor, breast mass localization, recognition of pathology (organ parts), infectious diseases, and coronary artery stenosis classification. CNN and Auto-encoder have been commonly implemented to solve challenging medical image problems. This is because the structure of these deep learning methods makes it possible to learn salient features from the data to create different levels of abstraction to achieve the required result.

In recent time, exploration of machine learning algorithms in detecting thoracic diseases has gained attention of medical image classification. Varshni et al. (2019) [1] discuss about Pneumonia detection using CNN based feature extraction. They appraise the functionality of pre-trained

CNN models utilized as feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays, then analytically determine the optimal CNN model for the classification. Prior to this work, no literature was found to perform the studies on the combination of CNN based feature extractions and supervised classifier algorithms for the underlying task. In the process of meliorating the model performance, they found that the customized model outperforms the results documented in the work of Benjamin Antin et al. [4] for the same problem of pneumonia detection. Although the results in [3] were overwhelming, there were still some limitations in the model [3] which are vital to keep in consideration. There is no history of the associated patient considered in the evaluation model. Secondly, only frontal chest X-rays were used but it has been shown that lateral view chest X-rays are also helpful in diagnosis [10]. Thirdly, since the model exercises a lot of convolutional layers, it has a high running time.

The table 1 is a comparison of pneumonia detection results.

Author	**Technique Used**	**AOC**
B. Antin [4]	**Logistic Regression**	**0.6**
B. Antin [4]	**DenseNet-121**	**0.609**
Varshni [1]	**DenseNet-169, SVM classifier**	**0.8**

Table 1: Comparing results of [1] and [4]

Rajpurkar et al. (2017) [3] develop an algorithm *CheXNet* that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists. CheXNet is a 121-layer CNN trained on ChestX-ray14 that was at the time the largest publicly available chest X-ray dataset, containing over 100,000 frontal-view X-ray images with 14 diseases. They found that CheXNet exceeds average radiologist performance on the F1 metric, and then extend CheXNet to detect all 14 diseases in ChestX-ray14.

Lakhani and Sundaram (2017) [5] proposed a method of detecting pulmonary tuberculosis following the architecture of two different Dense Convolutional Neural Networks (DCNNs) AlexNet and GoogleNet. Lung nodule classification mainly for diagnosing lung cancer proposed by Huang et al. [6] also adopted deep learning techniques. Performance of different variants of CNNs for abnormality detection in chest X-Rays was proposed by Islam et al. [7] using the publicly available OpenI dataset [8]. For the better exploration of machine learning in chest screening, Wang et al. (2017) [9] released a larger dataset of frontal chest X-Rays. Pranav Rajpurkar, Jeremy Irvin, et al. (2017) [3] explored this dataset for detecting pneumonia at a level better than radiologists, they referred their model as ChexNet which uses DenseNet-121 layer architecture for detecting all the 14 diseases from a lot of 112,200 images available in the dataset. After the CheXNet [3] model, Benjamin Antin et al.(2017) [4] worked on the same dataset and proposed a logistic regression model for detecting pneumonia. Pulkit Kumar, Monika Grewal (2017) [11] using the cascading convolutional networks contributed their research for multilabel classification of thoracic diseases. Zhe Li (2018) [12] recently proposed a convolutional network model for disease identification and localization.

Das, Ghosh et al. (2013) [2] address the development of computer assisted malaria parasite characterization and classification using machine learning approach based on light microscopic images of peripheral blood smears. They devised a feature selection and classification scheme by combining F-statistic, statistical learning techniques : Bayesian learning and support vector machine (SVM) in order to provide the higher classification accuracy using best set of discriminating features. They went on to compare the performance of these two classifiers under feature selection framework toward malaria parasite classification.

Rajpurkar et al. (2017) [3] develop an algorithm *CheXNet* that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists. CheXNet is a 121-layer CNN trained on ChestX-ray14 that was at the time the largest publicly available chest X-ray dataset, containing over 100,000 frontal-view X-ray images with 14 diseases. They found that CheXNet exceeds average radiologist performance on the F1 metric, and then extend CheXNet to detect all 14 diseases in ChestX-ray14.

Antin et al. (2017) [4] follow the approach of CheXNet to develop the algorithm *DenseNet* using a 121-layer *Dense* CNN.

The table 2 is a comparison of results of chest diseases detection from [13], [14], [15].

Parameters	CNN	BPNN2	CpNN2	CNN + GIST [14]	VGG16 [15]	VGG19 [15]
No. of images	120,120	1000	1000	637	8100	8100
Accuracy	92.40%	80.04%	89.57%	92%	86%	92%

Table 2: Comparing Results Across Papers

Abiyev and Ma'aitah (2018) [13] also design a CNN is designed for diagnosis of chest diseases. For comparative analysis, backpropagation neural network (BPNN) and competitive neural network (CpNN) are carried out for the classification of the chest X-ray diseases. The designed CNN, BPNN, and CpNN were trained and tested using the chest X-ray images containing different diseases. Several experiments were carried out through training of these networks using different learning parameters and a number of iterations. In both backpropagation and competitive networks, it was observed that the input image of size 3232 pixels showed good performance and achieved high recognition rates. Based on recognition rates, the backpropagation networks outperformed the competitive networks. Moreover, the competitive networks did not require manual labelling of training data as it was carried out for the backpropagation network. Furthermore, a CNN was also trained and tested using a larger dataset which was also used for training and testing of BPNN and CpNN. After convergence, it was noticed that the CNN was capable of gaining a better generalization power than that achieved by BPNN and CpNN, although required computation time and the number of iterations were roughly higher. This outperformance is mainly due to the deep structure of CNN that uses the power of extracting different level features, which resulted in a better generalization capability. The simulation result of proposed CNN is also compared with other deep CNN models such as GIST, VGG16, and VGG19. These networks have lower generalization capabilities and accuracies compared to the proposed network. The obtained results have demonstrated the high recognition rates of the proposed CNN.

4 Data Setup

The dataset for the chest X-ray images is taken from a Kaggle -Database. The dataset comprised 3 folders of chest X-rays each from COVID-19 patients, Normal Individuals and Pneumonia patients. All images were 1024X1024 pixels. The respective number of samples in this dataset are as shown in Table 3.

Patient Condition	**No. of cases**
COVID-19	**219**
Normal	**1341**
Pneumonia	**1345**

Table 3: Dataset

We first have to introduce a quantized desired response d(n) to the problem. Although the data we are using is for three classes, we are focusing on the two label classification to begin with, namely ’normal’ and ’covid-19’. The way we label the data for the classification is as follows:

$$d(n) = \begin{cases} 1, & \text{if the detected case is COVID-19} \\ 0, & \text{if the patient is healthy and normal} \end{cases}$$

There were variations in the colours of the images in this dataset so all of them are converted to grey scale images to maintain uniformity across the dataset. Also, the size of each image is reduced to 64X64. The CV2 (OpenCV) library in Python was used for processing the images to appropriate size and grayscale.

Any input to a perceptron, decision tree or Feed Forward Neural Network is in the form of array of numbers. Therefore a row of pixel value is calculated for each image in the dataset. All such rows are appended to create a text file with the rows representing each image in the dataset and columns corresponds to pixel values. Thus there were 4096 columns (64X64). A last column depicting the label of type of disease is added in the end as discussed above.

5 Proposed Models

We begin by implementing Perceptron and Decision Tree algorithms for the classification problem at hand and use the performance of these models as a benchmark to compare the feed forward neural network(FFNN) and convolutional neural network models (CNN) we further develop.

Using an appropriate metric to compare the performance of our models is also important. The classification problem we have is that of an unbalanced dataset, meaning, there is an imbalanced distribution of classes. In such cases, accuracy as a metric fails to provide better understanding of how the model is performing. Also, in our application, we want to reduce the number of false negatives and false positives as it is important to not wrongly classify normal as well as Covid-19 X-rays. F1-score can be used as a performance metric, when the False Negatives and False Positives are crucial. It is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the Accuracy Metric. More over, the literature on medical image classification uses F1-Score as a standard metric for model comparison. In our subsequent discussion, we provide both accuracy and F1-Scores to compare different models.

$$F1 - Score = \left(\frac{Recall^{-1} + Precision^{-1}}{2}\right)^{-1} = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

$$where$$

$$Precision = \left(\frac{True\,Positive}{True\,Positive\;+\;False\,Positive}\right),\; Recall = \left(\frac{True\,Positive}{True\,Positive\;+\;False\,Negative}\right)$$

5.1 Feed Forward Neural Network

Neural Networks which are inspired from biological learning system provide a robust approach to develop functional approximation of more complex and non-linear relationships between the features and labels. The hidden layers in these networks provide a non linear transformation and act as intermediate representation of transformed feature space. In our implementation of

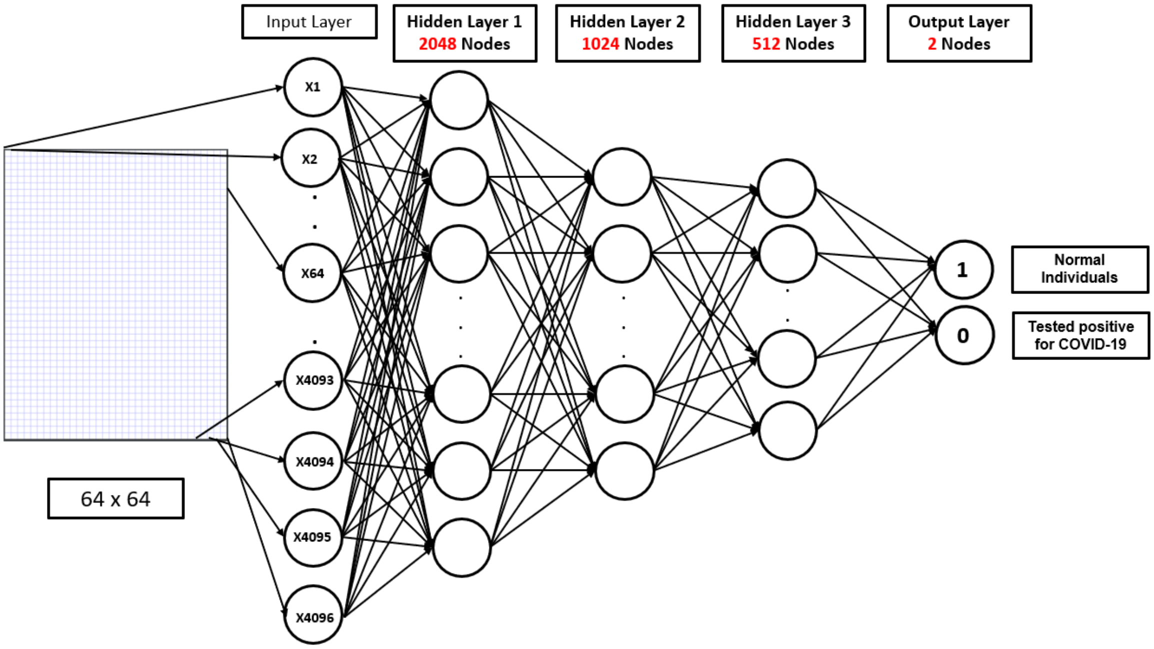


Figure 1: Feed Forward Neural Network - Architecture

FFNN, we build a network with an input layer which has 4096 nodes to take the data from 64x64 image as input followed by hidden layer 1 which has 2048 nodes, hidden layer 2 which has 1024 nodes, hidden layer 3 which has 512 nodes and a final out put layer with two output nodes. Figure 1 shows the architecture of our FFNN implementation.We can tune the performance of these networks by changing different hyper-parameters related to the model such as the optimizer used to solve the back propagation, batch size and epochs used for learning etc., we tune these parameters and present a comparative analysis of model performance in subsequent sections.

5.2 Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant

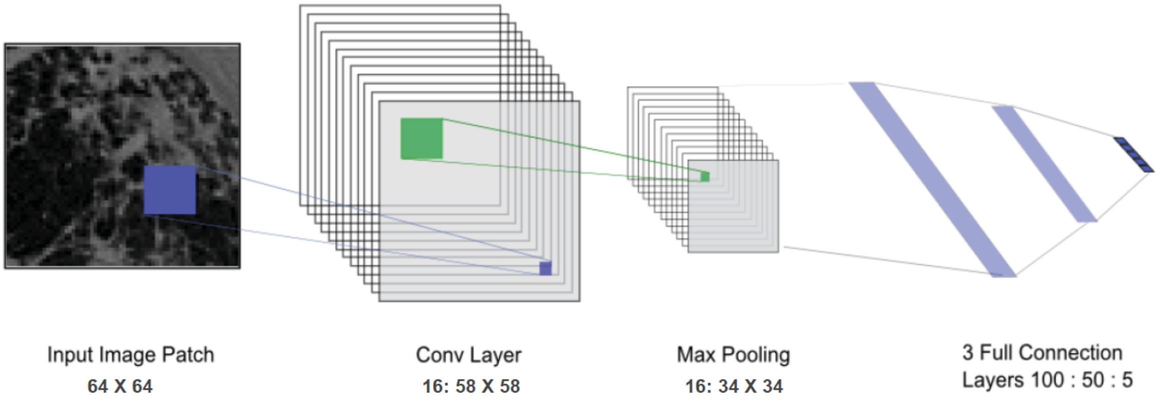


Figure 2: Convolutional Neural Network - Architecture

filters.

The first stage is a Convolutional Layer. The objective is to do a Convolution Operation to extract the high-level features such as edges, from the input image. Next is the Pooling layer which is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Now the input image is converted into a suitable form (consisting of dominant features),next the image is flattened into a column vector and trained using the Multi-Level Perceptron. This stage is known as the fully connected layer. Lastly a softmax technique can be used to classify the image into required classes.

We follow Li et. al (2014) in developing our CNN implementation. The overall framework design is illustrated in Fig. 2. Input of the network is normalized X-ray image . The first layer is a convolutional layer with kernel size of 7X7 pixels and 16 output channels. The second layer is a max pooling layer with 22 kernel size. The following three layers are fully connected neural layers with 100-50-5 neurons in each layer. We followed the work of Qing Li, Weidong Cai, Xiaogang Wang†, Yun Zhou‡, David Dagan Feng and Mei Chen (2014) [20]

	Perceptron	**Decision Tree**	**FFNN**	**CNN**
Accuracy	**95.23%**	**93.82%**	**97.75%**	**98.20%**
F1 Score	**0.834**	**0.761**	**0.98**	**0.989**

Table 4: Performance Comparision

6 Model Performance Optimization

6.1 Data and Hyper-parameters

To implement our models, we split the data into 75% training sample and 25% test sample and we use 20% of the training sample for validation, this makes the overall split as 60% training, 15% validation and 25% test sample. We use stratified sub-sampling to maintain the same distribution of labels across all subsample which is same as that of the total sample. We use an input image of size 64x64, batch size of 128 instances over 20 epochs. We use ”Categorical Cross Entropy” loss as the loss function to optimize using ”Adam” optimizer and ”Relu” activation functions. Unless specified otherwise, the results presented are based on these parameters.

6.2 Performance Comparison

Table 4 shows the model performance of perceptron, decision tree, FFNN and CNN. Results show that the basic models (perceptron and decision tree) perform relatively well on accuracy however they perform relatively poor on F1-score which is our metric of consideration. Across these two models, perceptron outperforms decision tree on F1-Score as well as accuracy.

Both FFNN and CNN perform better than the basics models on accuracy and show a drastic improvement in F1-score (0.83 to 0.98 and 0.99). This shows that the deep learning algorithms are a better solution for medical image classification. CNN actually performs slightly better than FFNN on accuracy (98.42% vs 97.75%) and F1-Score (0.99 vs 0.98).

6.3 McNemar’s Test

McNemar’s test which is also referred to as ”within-subjects chi-squared test” is a test used to compare different model performances based on a slightly modified version of confusion matrix. In a regular confusion matrix, we compare the model predictions with true labels. However, in the case of McNemar’s, we build a matrix comparing the predictions from two models. The Chi-Squared test statistic is computed based on the misclassifactions across both models.

We compare the FFNN and CNN models using McNemar’s test. Table 5 shows the confusion matrix comparing FFNN and CNN, diagonal elements shows that both the models are in agreement with regards to the classified labels. The off diagonal elements are used to calculate the Chi-Squared statistic. Table 6 shows the results for the McNemar’s test. The p-value of 0.0027 indicates that we can reject the Null Hypothesis that both models are equal.

Method		CNN		
		Covid-19	Normal	
FFNN	Covid-19	48	0	48
	Normal	9	388	397
	Total	57	388	445

Table 5: McNeamar’s Confusion Matrix

McNemar’s Test Results	
McNemar’s Chi-square:	9.00
p-value:	0.0027
Cramer’s phi:	0.1422

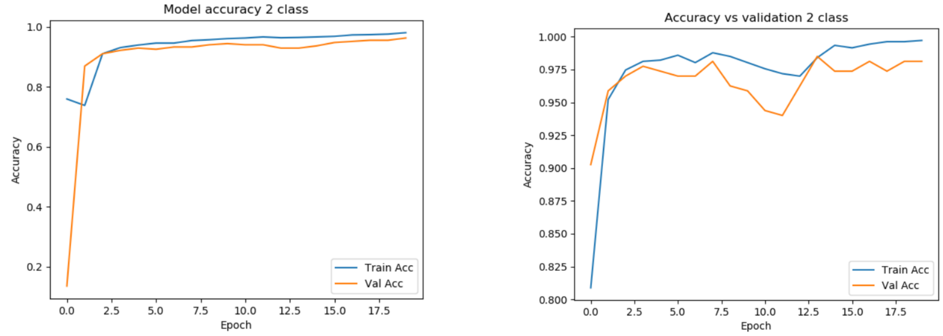
Table 6: McNeamar’s Test Results

6.4 Accuracy vs Epochs

In general increasing the epoch size causes the model to over-fit, as the weights keep on converging more on the training set. Also a too low epoch size means a under-fitting scenario where the model is not properly trained. The affect of increasing epoch size can be best studied by introducing validation set. Checking the accuracy on validation set prevents the model from over-fitting. Also the validation accuracy helps in fine tuning the model on the training set.

When the epoch value is too high the validation accuracy will go on decreasing or almost remains the same. Thus a high epoch value should be avoided. A low epoch value means that the model is not properly trained. Thus we observed that a medium epoch value between 10-15 fine tunes our model without over-fitting. A further increase in epoch doesn't have a much affect on accuracy. Thus choosing an appropriate medium epoch value will prevent over-fit and will also be efficient in terms of timing computation.

We iterate over the number of epochs and plot how the train and validation accuracy changes as the number of epochs increase. Figure 3 shows the plots of both FFNN and CCNN models. The models achieve a high validation accuracy around 10-15 epochs. Based on the validation accuracy result, we can conclude that a medium epoch value will prevent the problem of over-fitting.



(a) FFNN 2 class

(b) CNN 2 class

Figure 3: Accuracy vs Epoch Training and Validation Set

6.5 Image Size Comparison

The original image size is 1024X1024. Operating the model on this size will consume a lot of time. Scaling of image to a lower size may lead to loss of features. We decided to scale the image to 256X256 and 64X64 and compared the result on both the scales. Table 7 shows the comparison. The results were almost similar. Only the time taken to run the 256X256 image model is significantly higher. The results were enough to motivate us to operate on 64X64 images without the doubt of loss of image features.

	SIZE 64X64	**SIZE 256X256**
Accuracy	**98.20%**	**98.71%**
F1 Score	**0.989**	**0.992**
Average Time	**45 sec**	**∼16 mins**

Table 7: Model Comparison: CNN with Different Image Sizes

6.6 Data Normalization

The data normalization is necessary to ease the model learning problem. Without the normalization, gradients may end up taking a long time and can oscillate back and forth and take a long time before it can finally find its way to the global/local minimum. The range of pixel values in the given images is between 0-255. Hence we normalized the image by dividing each pixel values with 255. The table 8 shows the result of normalized and not normalized dataset with 2 class classification.The normalized model consistently performed better compared to the non-normalized one.

	Without Normalization		**With Normalization**	
	FFNN	**CNN**	**FFNN**	**CNN**
Accuracy	**96.17%**	**96.45%**	**97.75%**	**98.20%**
F1 Score	**0.975**	**0.979**	**0.98**	**0.989**

Table 8: Model Comparison: With Normalization vs Without Normalization

6.7 Image Orientation and Data Augmentation

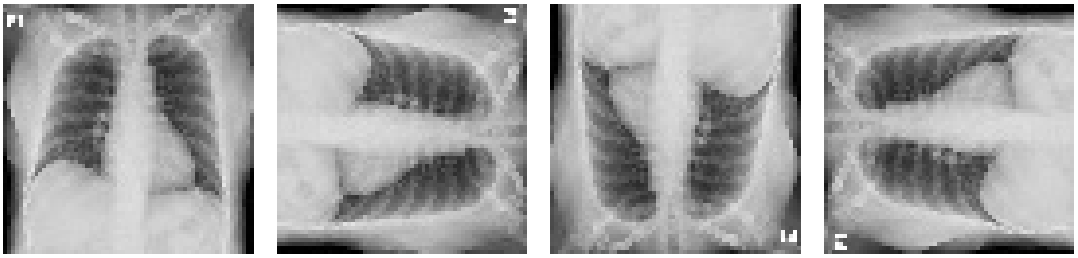


Figure 4: Rotated image dataset

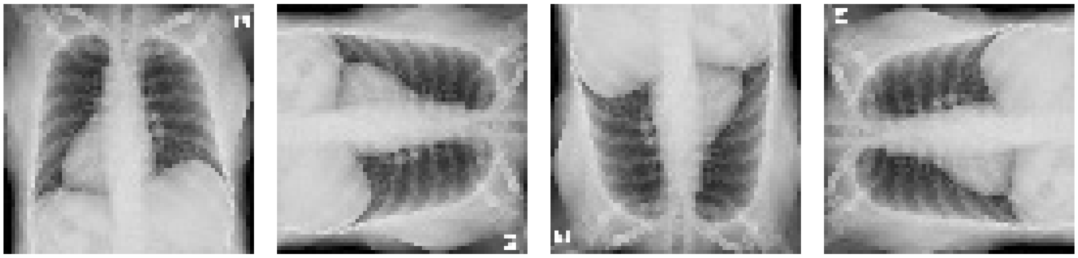


Figure 5: Mirrored image dataset

COVID-19 shows presence of ”cough” like cluster in the lungs. This blob can be present anywhere inside the lungs and in any shape. Any radiologist identifies this shape from the X-ray images and get to a conclusion. A CNN does the same thing i.e. extracting this feature. But since the shape, size and location is not fixed of this blob, this can be associated with the viewpoint problem.

We only have limited number of images. This will not capture all type of possible information like rotation as seen in Figure 4.Therefore we include these rotated images as part of our dataset and run the models again. Further, since the lungs are vaguely symmetric over the vertical axis (along the sternum), we include the corresponding mirror images of the entire dataset as seen in Figure 5. Hence we mirrored the images to train our model to see any changes in accuracy and the F1 score.

Since CNN operates on the features, rather than the location or size, we expected the results to be more or less similar to one observed in normal dataset. On the other hand, FFNN operates on data matrix and its training model will be affected by new set of datasets. And as per the tables 9 and 10, the results are not consistent for all the cases. We can see that there is an improvement in the accuracy and the F1 score for CNN using augmented data set but in the case of FFNN, some optimizers seem to be performing better that the rest. More is discussed in the next section.

6.8 Comparison of Different Optimizers

Optimizers update the weight parameters to minimize the loss function. Loss function acts as guides to the terrain telling optimizer if it is moving in the right direction to reach the bottom of the valley, the global minimum. We adopted 3 optimizer methods to check the performance - Stochastic Gradient Descent, RMS Prop, Adam. SGD is the simplest approach where we update parameters in the negative gradient direction to minimize the loss. The learning rate is constant. RMSProp is Root Mean Square Propagation. It uses moving average of the squared gradient. In RMSProp learning rate gets adjusted automatically and it chooses a different learning rate for each parameter. Adam calculates the individual adaptive learning rate for each parameter from estimates of first and second moments of the gradients. It is one of the most efficient optimizers.

From the table 9 we see that Adam optimizers performed better or almost similar to other optimizers. This justifies why adam is considered one of the most efficient optimizers. We then compare the results of these optimizers on the augmented data set and observe that RMSprop gives us the best accuracy for CNN and ADAM gives us the best overall accuracy and F1 score for an FFNN. Results as seen in table 10

	SGD		**RMSprop**		**ADAM**	
	FFNN	**CNN**	**FFNN**	**CNN**	**FFNN**	**CNN**
Accuracy	**97.75%**	**96.40%**	**96.17%**	**98.20%**	**97.75%**	**98.20%**
F1 Score	**0.988**	**0.979**	**0.975**	**0.989**	**0.98**	**0.989**

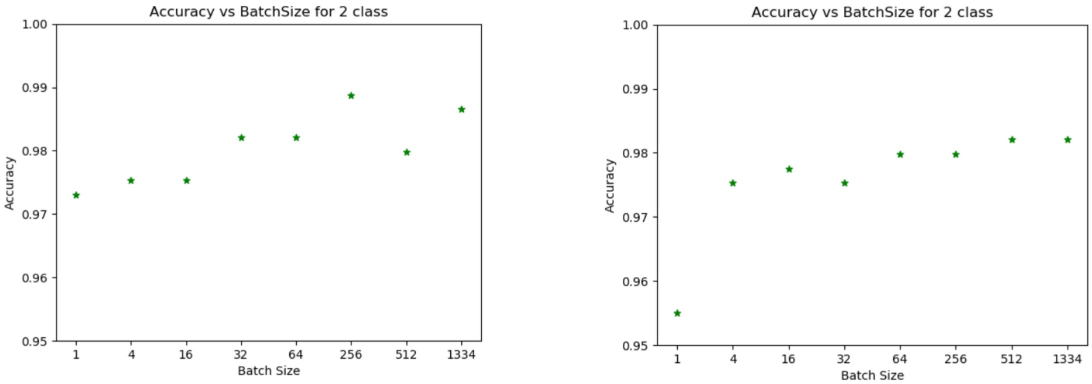
Table 9: Model Comparison with Different Optimizers

	SGD		RMSprop		ADAM	
	FFNN	CNN	FFNN	CNN	FFNN	CNN
Accuracy	**96.417%**	**96.833%**	**95.425%**	**98.72%**	**98.4005%**	**97.98%**
F1 Score	**0.9789**	**0.981**	**0.9729**	**0.993**	**0.9906**	**0.988**

Table 10: Model Comparison with Different Optimizers on Augmented data set

6.9 Accuracy vs Batch Size

Updation of weights after the gradient calculation depends on the batch size. That is batch size will tell how many instances from the dataset has to be checked before calculating the error gradient and thus modifying the weights. When weights are updated too frequently, the model takes into consideration the noisy data. A high batch size will cause the dataset to average out and ignoring some of the crucial data points. Therefore a medium batch size is perfect which will eliminate shortcomings of too low and too high batch size. The results in the figure 6 shows that medium epoch value performed always better.



(a) FFNN

(b) CNN

Figure 6: Accuracy vs Batch size - 2 Class Classification

6.10 HSV Filtering

In order to get better results from the image dataset, we tried to use some filtering techniques that would enhance the features of the images. Based on the work of Xin Zhou, Miaofei Han, Yanli Song, Qiang Li[18], we tried an HSV (Hue, Saturation , and Value) filtering as a data pre processing technique for colour space decomposition along with a 0-255 normalization on each channel. We developed a tool in python to apply different levels of this filter to determine the best setting for our purpose.

In the end, we determined that the images we got after this filtering were not extremely useful since here we are dealing with a gradient of the pixels to determine if there is any abnormality. This technique would focus more on hard edges like bones as seen in Anil K. Bharodiya , Atul M. Gonsai [19] instead of soft organs hence would only go with normalization across the channels.

7 COVID-19 Diagnosis Web Application

To supplement our work with the COVID-19 diagnosis machine learning model, we have developed a web application, as seen in Fig: 7, using Flask and programmed in Python. It works hand-in-hand with our neural network model. A test X-ray scan can be uploaded on this application as a normal document upload to the web, refer Fig: 8, and with our saved model architecture and trained network weights running in the background, it can accurately display a prediction, whether the test scan is from a 'NORMAL' individual, as seen in Fig. 10 or if it is tested 'COVID-19 POSITIVE', as seen in Fig. 9. Our neural network model can take an image scan of any size or colour noise, since the pre-processing phase of our model adjusts the size and gray-scaling as suitable for our model.

This tool can be further developed for testing batches of X-ray scans, which can significantly increase the capacity of preliminary testing and save on analysis times at the diagnostic centers.

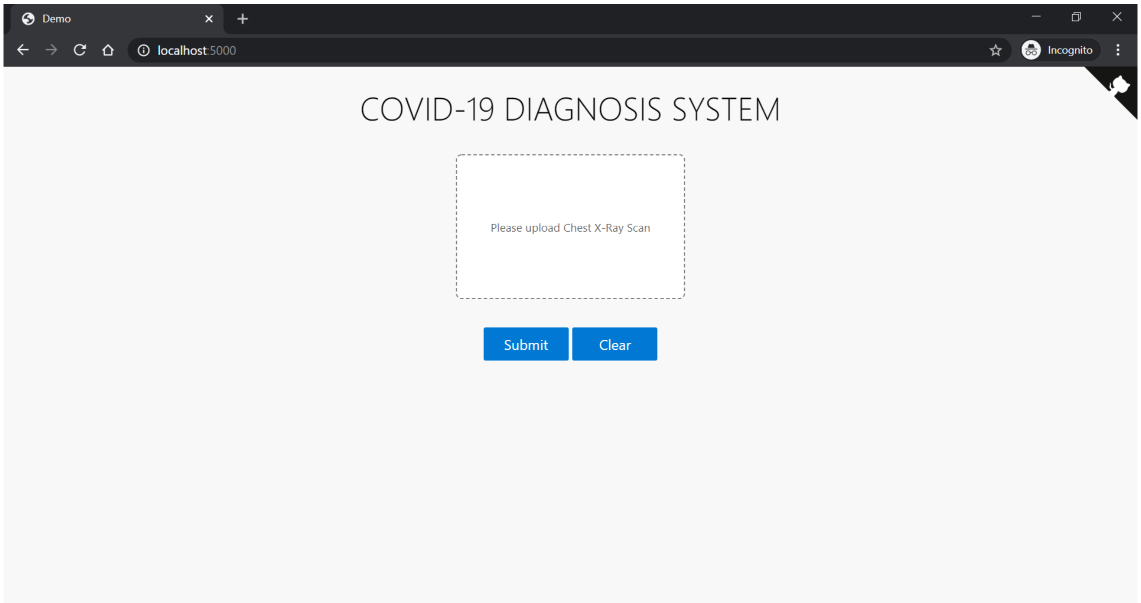


Figure 7: Homescreen of the Web Application

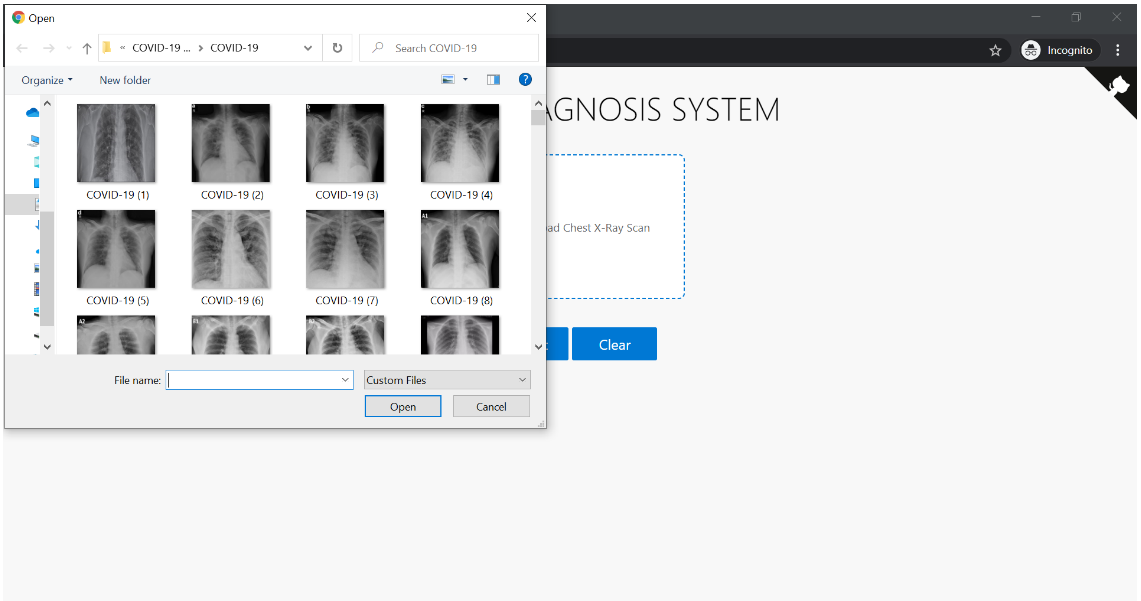


Figure 8: Uploading test X-ray for prediction

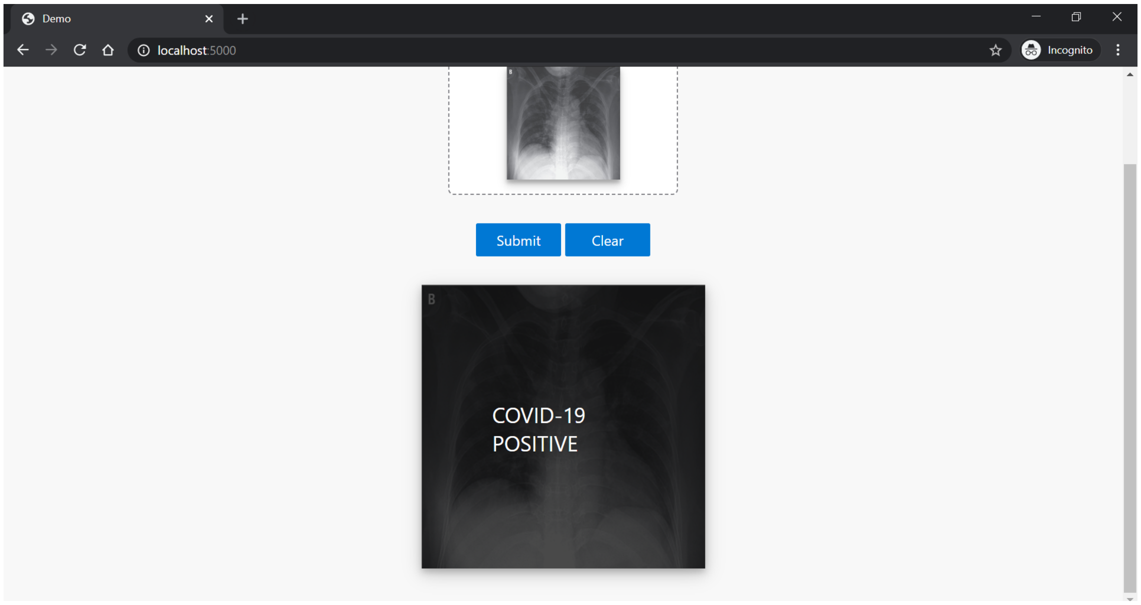


Figure 9: An X-Ray tested POSITIVE for COVID-19

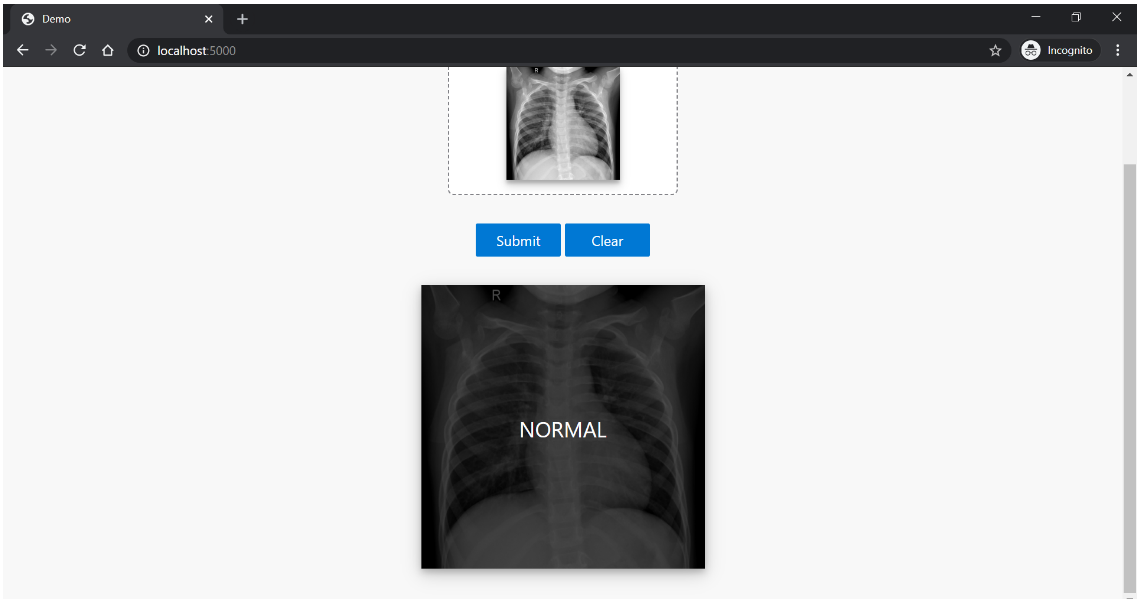


Figure 10: An X-Ray tested NORMAL

Since it is a novel virus, very few domain experts in radiology have adequate knowledge about the specific characteristics of this virus. If the neural network is trained on a larger data-set of patient X-ray scans, the network will extract additional features which can be displayed as insights in this tool. Thus, it can also be used for analysis of the virus attack.

8 Three Class Classification: Covid-19, Normal and Pneumonia

We replicated the above discussed approaches and model for a three class classification problem where we included the image data from Pneumonia patients to the dataset. In this section, we report the results for those models. Here, d(n) is the quantized desired response for classification.

$$d(n) = \begin{cases} 1, & \text{if the detected case is COVID-19} \\ 0, & \text{if the patient is healthy and normal} \\ 2, & \text{if the detected case is pneumonia} \end{cases}$$

8.1 Performance Comparision

Table 11 show the accuracy and F1-score for CNN and FFNN in a 3-class classification problem. The F1-scores are relatively low compared to the 2-class classification problem

	FFNN	**CNN**
Accuracy	**92.57%**	**96.15%**
F1 Score	**0.828**	**0.861**

Table 11: Performance Comparison - 3 Class Problem

8.2 McNemar’s Test

Table 12 and table 13 show the results for McNemar’s Test for a 3-Class classification problem. P-value of 0.7055 indicate that we can not reject the Null Hypothesis that both models are equal.

Method		CNN			
		Covid-19	Normal	Pneumonia	Total
FFNN	Covid-19	38	4	10	52
	Normal	3	366	65	434
	Pneumonia	2	11	282	295
	Total	43	381	357	781

Table 12: McNeamar’s Confusion Matrix - 3 Class Problem

McNemar’s Test Results	
McNemar’s Chi-square:	0.1429
p-value:	0.7055
Cramer’s phi:	0.0096

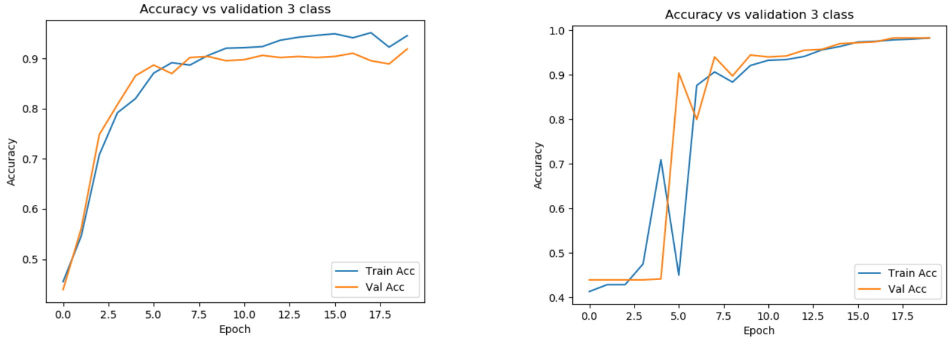
Table 13: McNeamar’s Test Results - 3 Class Classification

8.3 Accuracy vs Epochs

Figure 11 shows the change in accuracy on training and validation set over different epochs. For 3 class classification using CNN, maximum accuracy on validation sample is at around 20 epochs.

8.4 Data Normalisation

Table 14 shows the results for data with and with out normalization on a three class classification problem. Models using normalized data outperform the models using regular data even in three class classification.



(a) FFNN 3 class

(b) CNN 3 class

Figure 11: Accuracy vs Epoch Training and Validation Set

	Without Normalization		**With Normalization**	
	FFNN	**CNN**	**FFNN**	**CNN**
Accuracy	**84.89%**	**94.38%**	**92.57%**	**96.15%**
F1 Score	**0.567**	**0.888**	**0.828**	**0.914**

Table 14: Model Comparison: With Normalization vs Without Normalization - 3 Class

8.5 Comparison of Different Optimizers for Three Class Classification

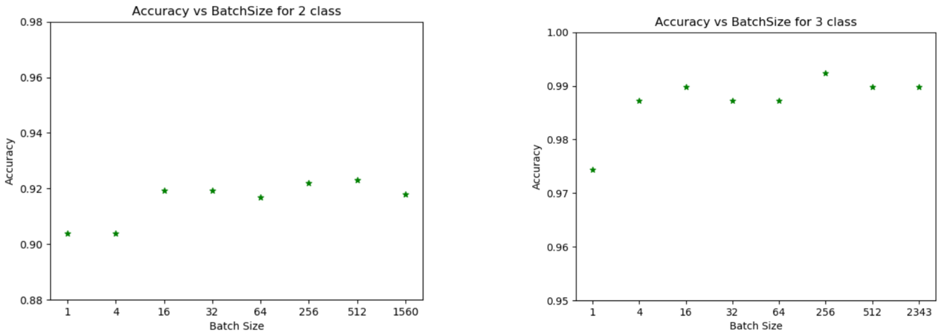
As expected, Adam performs better both on Accuracy and F1-Score in FFNN and CNN.

	SGD		**RMSprop**		**ADAM**	
	FFNN	**CNN**	**FFNN**	**CNN**	**FFNN**	**CNN**
Accuracy	**88.73%**	**75.92%**	**87.06%**	**93.34%**	**92.57%**	**96.15%**
F1 Score	**0.804**	**0.778**	**0.766**	**0.851**	**0.828**	**0.914**

Table 15: Model Comparison with Different Optimizers - 3 Class Classification

8.6 Accuracy vs Batch Size

Figure 12 show the comparison for three class classification problem. We can see that medium batch sizes are better than too low or too high batch sizes.



(a) FFNN

(b) CNN

Figure 12: Accuracy vs Batch size - 3 Class Classification

9 Future Scope

Since an X-ray diagnosis is a low specificity test, a possible future enhancement of this work would be the implementation of an ensemble learner, that can take in auxiliary features about the patient such as travel history, medical history, age, etc. besides a chest X-ray scan. This will help increase the specificity of the diagnosis as cases diagnosed with a recent travel across international boundaries, aged above 50 yrs and with past history of respiratory syndromes, were higher in number ($\geq 75\%$).

Further, since a recurrent neural network can work best with temporal data, a potential artificial intelligence technique for analysis of the virus could be tracking the growth of the coronavirus inside a patient's body, with X-ray scans taken over a period of time. This can assist tracing the attack of the virus from the onset of the disease, and yield some novel insights about the infection inside the body.

As we have successfully built neural network model for a 3 class classification problem, we propose multi-class classification models based on similar neural network architecture for predicting the diagnosis of other respiratory infectious diseases that can also be diagnosed using a chest X-ray scan.

10 Conclusion

To sum it all, machine learning approaches, especially convolutional neural networks show a real promise in diagnosing the novel coronavirus using chest X-rays. We have proved this while comparing across the traditional learners, which yield significantly low F1 scores as compared to a neural network. The optimization techniques implemented have improved the performance of our model. A larger database of COVID-19 patient's X-rays can help in the further generalization of the model and remove bias. Although, this work cannot completely replace traditional radiology diagnosis test, it can certainly supplement the diagnostic centers across the globe.

References

[1] Varshni, D.; Thakral, K.; Agarwal, L.; Nijhawan, R.; Mittal, A. Pneumonia Detection Using CNN based Feature Extraction. In Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT); 2019; pp. 1–7.

[2] Das DK, Ghosh M, Pal M, Maiti AK, Chakraborty C., "Machine learning approach for automated screening of malaria parasite using light microscopic images", Micron 45 (2013), 97106, 2013.

[3] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. Lungren and A. Ng, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning", arXiv:1711.05225, Dec 2017.

[4] Benjamin Antin, Joshua Kravitz, Emil Martayan, "Detecting Pneumonia in Chest X-Rays with Supervised Learning", semanticscholar.org, 2017.

[5] Paras Lakhani and Baskaran Sundaram. 2017. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology 284, 2 (2017), 574582.

[6] Kai-Lung Hua, Che-Hao Hsu, Shintami Chusnul Hidayati, Wen-Huang Cheng, and Yu-Jen Chen. 2015. Computer-aided classification of lung nodules on computed tomography images via deep learning technique.OncoTargets and therapy 8 (2015).

[7] Mohammad Tariqul Islam, Md Abdul Aowal, Ahmed Tahseen Minhaz, and Khalid Ashraf. 2017. Abnormality detection and localization in chest x-rays using deep convolutional neural networks. arXiv:1705.09850 (2017).

[8] 2016. Openi Dataset :. (2016). https://openi.nlm.nih.gov

[9] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri and Ronald M Summers. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 34623471.

[10] Suhail Raoof, David Feigin, Arthur Sung, Sabiha Raoof, Lavanya Irugulpati, and Edward C Rosenow III. 2012. Interpretation of plain chest roentgenogram. Chest 141, 2 (2012), 545558.

[11] Pulkit Kumar, Monika Grewal, and Muktabh Mayank Srivastava. 2018. Boosted cascaded convnets for multilabel classification of thoracic diseases in chest radiographs. In International Conference Image Analysis and Recognition. Springer, 546552.

[12] Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li, and F Li. 2017. Thoracic disease identification and localization with limited supervision. arXiv preprint arXiv:1711.06373 (2017).

[13] Rahib H. Abiyev, Mohammad Khaleel Sallam Ma'aitah. Deep Convolutional Neural Networks for Chest Diseases Detection (2018)

[14] Bar Y., Diamant I., Wolf L., Lieberman S., Konen E., Greenspan H. Chest pathology detection using deep learning with non-medical training. Proceedings of Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium; 2015; Brookly, NY, USA. pp. 294–297.

[15] Islam M. T., Aowal M. A., Minhaz A. T., Ashraf K. Abnormality detection and localization in chest x-rays using deep convolutional neural networks. 2017. Arxiv.

[16] Russakovsky O., Deng J., Su H., et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision. 2015;115(3):211–252. doi: 10.1007/s11263-015-0816-y.

[17] Tobore, I., Li, J., Yuhang, L., Al-Handarish, Y., Kandwal, A., Nie, Z., & Wang, L. (2019). Deep Learning Intervention for Health Care Challenges: Some Biomedical Domain Considerations. JMIR mHealth and uHealth, 7(8), e11966. https://doi.org/10.2196/11966

[18] Xin Zhou, Miaofei Han, Yanli Song, Qiang Li. Fast filtering techniques in medical image classification and retrieval, Medical Image Information Laboratory, Advanced Medical Equipment Research Center, Shanghai Advanced Research Institute, Chinese Academy of Sciences, 99 Haike Road, Building No. 3, Pudong, Shanghai 201210, China.

[19] Anil K. Bharodiya , Atul M. Gonsai, (2019). An improved edge detection algorithm for X-Ray images based on the statistical range, https://doi.org/10.1016/j.heliyon.2019.e02743

[20] Qing Li, Weidong Cai, Xiaogang Wang†, Yun Zhou‡, David Dagan Feng and Mei Chen (2014),Medical Image Classification with Convolutional Neural Network, 2014 13th International Conference on Control, Automation, Robotics Vision.

[21] Luís A. Bastião Silva Luís S. Ribeiro Milton Santos Nuno Neves Dulce Francisco Carlos Costa José Luis Oliveira (2015), Normalizing Heterogeneous Medical Imaging Data to Measure the Impact of Radiation Dose, Society for Imaging Informatics in Medicine 2015

Appendices

A Group Picture

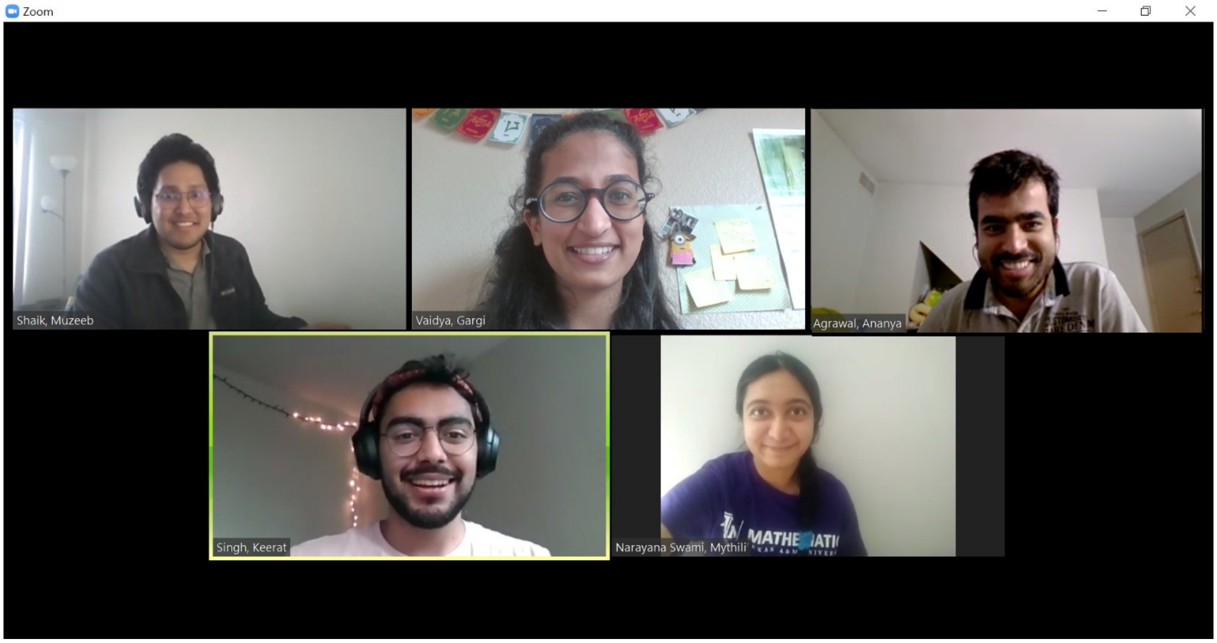


Figure 13: Our Zoom Meetings for Project Discussions

B Virtual Meetings and Workload Management

To manage the project better, we decided to make progress in small sprints and review regularly. To do this, we scheduled meetings every alternate day from the beginning to the submission deadline. Gargi is responsible for overall project management, Ananya is responsible for compiling the code from everyone and provide the final version of the code implementation to be submitted. Keerat helped every one with GitHub related issues and coordinated with Ananya for code compilation. Muzeeb is responsible for the write-up and coordinated with others to complete the write-up on OverLeaf. In terms of implementation, Ananya and Keerat are responsible for data preparation and

CNN implementation, Muzeeb is responsible for FFNN. Gargi worked on the WebApp implementation. Mythili did the investigation on different research papers and identified the best possible paper that can be referenced and implemented.